

LOWER BOUNDS ON THE GROWTH OF GRIGORCHUK'S TORSION GROUP

LAURENT BARTHOLDI

ABSTRACT. In 1980 Rostislav Grigorchuk constructed a group G of intermediate growth, and later obtained the following estimates on its growth γ [Gri85]:

$$e^{\sqrt{n}} \lesssim \gamma(n) \lesssim e^{n^\beta},$$

where $\beta = \log_{32}(31) \approx 0.991$. He conjectured that the lower bound is actually tight.

In this paper we improve the lower bound to

$$e^{n^\alpha} \lesssim \gamma(n),$$

where $\alpha \approx 0.5157$, with the aid of a computer. This disproves the conjecture that the lower bound be tight.

1. INTRODUCTION

The growth of finitely generated groups, in relation with properties of differentiable manifolds and coverings, was studied since the 1950's in the former USSR [Sva55] and in the 1960's in the West [Mil68a]: let the finitely generated group G be the fundamental group of a compact CW -complex K . Then the growth of G is equivalent to the growth of the universal cover \tilde{K} . There are well-known classes of groups of polynomial growth: abelian (K a torus, \tilde{K} Euclidean space), and more generally virtually nilpotent groups [Gro81]; and classes of exponential growth: non-virtually-nilpotent linear [Tit72] or non-elementary hyperbolic [GH90] groups (K a negatively curved complex). John Milnor asked whether there exist finitely generated groups with growth greater than polynomial but less than exponential [Mil68b]. The first example of such a group, of intermediate growth, was discovered by Rostislav Grigorchuk, see [Gri83, Gri85, Gri91]. He showed that the growth γ of his group satisfies

$$e^{\sqrt{n}} \lesssim \gamma(n) \lesssim e^{n^\beta},$$

where $\beta = \log_{32}(31) \approx 0.991$. The author obtained the following improvement in [Bar98]: Let η be the real root of the polynomial $X^3 + X^2 + X - 2$, and set $\beta' = \log(2)/\log(2/\eta) \approx 0.767$. Then the growth γ of Grigorchuk's group satisfies

$$e^{\sqrt{n}} \lesssim \gamma(n) \lesssim e^{n^{\beta'}}.$$

Recently, the same result was rediscovered and extended to a larger class of groups by Roman Muchnik and Igor Pak [MP99].

A remaining outstanding problem in the theory of growth of groups is the question, raised by Grigorchuk, of the existence of groups with growth exactly $e^{\sqrt{n}}$. Such groups would have interesting extremal properties, for instance being of finite width [BG98]. He conjectured in [Gri85] that his group has this property; but in this paper we disprove that conjecture with the following result:

Date: February 1, 2008.

1991 *Mathematics Subject Classification.* **20F32** (Geometric group theory),
16P90 (Growth rate),
20E08 (Groups acting on trees),
05C25 (Graphs and groups)

This work has been supported by the "Swiss National Science Foundation".

Theorem 1. *The growth of Grigorchuk's 2-group G satisfies*

$$\gamma(n) \gtrsim e^{n^\alpha},$$

where $\alpha = 0.5157$.

However, the following construction could produce a group of growth $e^{\sqrt{n}}$: let G be Grigorchuk's 2-group, generated by $S = \{a, b, c, d\}$. Let \mathfrak{L} be the graded Lie 2-algebra of G (see [BG98]), and let \mathfrak{G} be the group generated by the $1 + (s - e)$, $s \in S$ in the enveloping algebra $\mathcal{U}(\mathfrak{L})$. There are interesting relations between the growth of G and that of \mathfrak{G} , and it seems the growth of \mathfrak{G} is "smoother".

The approach used in this paper to obtain lower bounds on the growth was suggested, with small variations, by Yuriĭ Leonov [Leo98], where he announced $\gamma(n) \gtrsim e^{n^\beta}$ for $\beta = \log_{87/22}(2) \cong 0.504$. I wish to thank Yuriĭ for introducing me to this question, and Slava Grigorchuk and Pierre de la Harpe for their interest.

2. GROWTH OF GROUPS

Let G be a group generated as a monoid by a finite set S . A *weight* on (G, S) is a function $\omega : S \rightarrow \mathbb{R}_+^*$. It induces a norm (again called a *weight*) ∂_ω on G by

$$\partial_\omega : \begin{cases} G \rightarrow \mathbb{R}_+ \\ g \mapsto \min\{\omega(s_1) + \dots + \omega(s_n) \mid s_1 \dots s_n =_G g\}. \end{cases}$$

The important properties of the weight ∂_ω are its *submultiplicativity*: $\partial_\omega(gh) \leq \partial_\omega(g) + \partial_\omega(h)$ and its *properness*: $|\{g \in G \mid \partial_\omega(g) \leq n\}| < \infty$ for all $n \in \mathbb{N}$.

A *minimal form* of $g \in G$ is a representation of g as a word of minimal weight over S . If a minimal form has been fixed, it will be denoted by \bar{g} . The *growth* of G with respect to ω is then

$$\gamma_\omega : \begin{cases} \mathbb{R}_+ \rightarrow \mathbb{R}_+ \\ n \mapsto \#\{g \in G \mid \partial_\omega(g) \leq n\}. \end{cases}$$

Alternatively, S can altogether be suppressed from the definition, and a weight can be defined as a function $\omega : G \rightarrow \mathbb{R}_+$ that has finite support generating G as a monoid.

The function $\gamma : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is *dominated* by $\delta : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, written $\gamma \lesssim \delta$, if there is a constant $C \in \mathbb{R}_+$ such that $\gamma(n) \leq \delta(Cn)$ for all $n \in \mathbb{R}_+$. Two functions $\gamma, \delta : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ are *equivalent*, written $\gamma \sim \delta$, if $\gamma \lesssim \delta$ and $\delta \lesssim \gamma$.

The following lemmata are well known:

Lemma 2. *Let S and S' be two finite generating sets for the group G , and let ω and ω' be weights on (G, S) and (G, S') respectively. Then $\gamma_\omega \sim \gamma_{\omega'}$.*

Proof. Without loss of generality we may suppose $1 \notin S$, so $\partial_\omega(s) \neq 0$ for any $s \in S$. Let $C = \max_{s \in S} \partial_{\omega'}(s)/\partial_\omega(s)$. Then $\partial_{\omega'}(g) \leq C\partial_\omega(g)$ for all $g \in G$, and thus $\gamma_\omega(n) \leq \gamma_{\omega'}(Cn)$, from which $\gamma_\omega \lesssim \gamma_{\omega'}$. The opposite relation holds by symmetry. \square

In particular, for fixed S , any weight ∂_ω is equivalent to the length-weight $|g| = \min\{n \mid g = s_1 \dots s_n, s_i \in S\}$.

The *growth type* of a finitely generated group G is the \sim -equivalence class containing its growth functions; it will be denoted by γ_G .

Note that all exponential functions b^n are equivalent, and polynomial functions of different degrees are inequivalent; the same holds for the subexponential functions e^{n^α} . We have

$$0 \lesssim n \lesssim n^2 \lesssim \dots \lesssim e^{n^\alpha} \lesssim e^{n^\beta} \lesssim \dots \lesssim e^n \quad \text{for } 0 < \alpha < \beta < 1.$$

Note also that the ordering \lesssim is not linear. Actually, Slava Grigorchuk showed in his pioneering paper [Gri85, Theorem 7.2] that \lesssim admits chains and antichains with the cardinality of the continuum.

Lemma 3. *Let G be a finitely generated group. Then $\gamma_G \lesssim e^n$.*

Proof. Choose for G a finite generating set S , and define the weight ω by $\omega(s) = 1$ for all $s \in S$. Then $\gamma_\omega(n) \leq |S|^n$ for all n , so $\gamma_G \lesssim e^n$. \square

If there is a $d \in \mathbb{N}$ such that $\gamma_G \lesssim n^d$, the group G is of *polynomial growth* of degree at most d ; if $\gamma_G \sim e^n$, then G is of *exponential growth*; otherwise G is of *intermediate growth*. The existence of groups of intermediate growth was first shown by Grigorchuk [Gri83].

Lemma 4. *Let $H < G$ be an index- N subgroup inclusion, let ω be a weight on G with growth function γ_ω , and let γ_ω^H denote the restriction of γ_ω to H :*

$$\gamma_\omega^H(n) = \#\{g \in H \mid \partial_\omega(g) \leq n\}.$$

Then there is a constant K such that

$$\gamma_\omega(n - K) \leq N\gamma_\omega^H(n) \leq \gamma_\omega(n + K)$$

holds for all n .

Proof. Let T be a transversal of H in G , and set $K = \max_{t \in T} \partial_\omega(t)$. For every $g \in G$ of weight at most $n - K$, there is a unique $t \in T$ with $gt \in H$, and $\partial_\omega(gt) \leq n$. The map $\{G \rightarrow H, g \mapsto gt\}$ is N -to-1, so its restriction to the set of elements of weight at most $n - K$ is at most N -to-1, proving the first inequality.

Considering all $h \in H$ of weight at most n and all $t \in T$, we have $N\gamma_\omega^H(n)$ distinct elements $ht \in G$, with $\partial_\omega(ht) \leq n + K$. This proves the second inequality. \square

3. THE GRIGORCHUK 2-GROUP

Let Σ^* be the set of finite sequences over $\Sigma = \{0, 1\}$. For $x \in \Sigma$ set $\bar{x} = 1 - x$. Define recursively the following length-preserving permutations of Σ^* :

$$\begin{aligned} a(x\sigma) &= \bar{x}\sigma; \\ b(0\sigma) &= 0a(\sigma), & b(1\sigma) &= 1c(\sigma); \\ c(0\sigma) &= 0a(\sigma), & c(1\sigma) &= 1d(\sigma); \\ d(0\sigma) &= 0\sigma, & d(1\sigma) &= 1b(\sigma). \end{aligned}$$

Then G , the Grigorchuk 2-group [Gri80, Gri85], is the group generated by $S = \{a, b, c, d\}$. It is readily checked that these generators are of order 2 and that $V = \{1, b, c, d\}$ is a Klein group.

Let $H = V^G$ be the normal closure of V in G . It is of index 2 in G and preserves the first letter of sequences; i.e. $H \cdot x\Sigma^* \subset x\Sigma^*$ for all $x \in \Sigma$. There is an injective homomorphism $\psi : H \rightarrow G \times G$, written $g \mapsto (g_0, g_1)$, defined by $0g_0(\sigma) = g(0\sigma)$ and $1g_1(\sigma) = g(1\sigma)$. As $H = \langle b, c, d, b^a, c^a, d^a \rangle$, we can write ψ explicitly as

$$\psi : \begin{cases} b \mapsto (a, c), & b^a \mapsto (c, a) \\ c \mapsto (a, d), & c^a \mapsto (d, a) \\ d \mapsto (1, b), & d^a \mapsto (b, 1). \end{cases}$$

Let B be the normal closure of $\langle b \rangle$ in G . We shall use the following facts, whose proof appears for instance in [Har, Section VIII.C]: B is of index 8 in G , and $\psi(H) = (B \times B) \rtimes \langle (a, 1), (d, 1) \rangle$ is of index 8 in $G \times G$, transversal to the order-8 dihedral group $\langle (a, d), (d, a) \rangle$.

4. THE GROWTH OF G

Definition 5. A weight ω on (G, S) is triangular if for any ordering (x, y, z) of $\{b, c, d\}$ one has

$$\omega(x) \leq \omega(y) + \omega(z).$$

We shall only consider triangular weights on G , and implicitly use the

Lemma 6. Let ω be a triangular weight. Then every $g \in G$ admits a minimal form

$$[*]a * a * a \cdots a[*],$$

where $*$ $\in \{b, c, d\}$ and the first and last $*$ s are optional.

Proof. Let w be a minimal form for $g \in G$. First, w may not contain two equal consecutive letters, since they could be cancelled, shortening the representation of g . Second, two unequal consecutive letters among $\{b, c, d\}$ can be replaced by the third, and as ω is triangular this operation will shorten the length of w while not enlarging its weight. Ultimately it will yield a word w' of the required form, also representing g , and with smaller or equal weight. It can then be chosen as a minimal form of g . \square

A lower estimate on the growth of G comes from the following proposition:

Proposition 7. There are a weight ω on S and constants $K \geq 0, \eta < 4$ such that for all $h \in H$, writing $\psi(h) = (h_0, h_1)$, we have

$$(1) \quad \partial_\omega(h) \leq \eta \max\{\partial_\omega(h_0), \partial_\omega(h_1)\} + K.$$

Corollary 8. The growth of G satisfies

$$\gamma_G \gtrsim e^{n^\alpha},$$

where $\alpha = \log(2)/\log(\eta) \gtrsim 0.5$.

Proof. Applying Lemma 4 (with constants K_1 and K_2) to the subgroup pairs $H < G$ and $\psi(H) < G \times G$, we obtain from the previous proposition

$$\frac{\gamma_\omega(\eta x + K + K_1)}{2} \geq \gamma_\omega^H(\eta x + K) \geq \gamma_{\omega \times \omega}^{\psi(H)}(x) \geq \frac{\gamma_\omega(x - K_2)^2}{8}$$

for all $x \geq K_2$. At the cost of increasing K , we rewrite it as $\gamma_\omega(\eta x + K) \geq \gamma_\omega(x)^2/4$, whence by iteration

$$\gamma_\omega\left(\eta^m x + \frac{\eta^m - 1}{\eta - 1}K\right) \geq \frac{\gamma_\omega(x)^{2^m}}{4^{2^m - 1}}$$

for all integers $m \geq 1$ and $x \in \mathbb{R}_+$. Let $L \in \mathbb{R}_+$ be such that $\gamma(L) > 4$. Now given $n \in \mathbb{R}_+$, let $m \in \mathbb{N}$ be maximal such that $x := \eta^{-m}(n - K(\eta^m - 1)/(\eta - 1))$ be at least equal to L . We then have $\gamma_\omega(n) \geq (\gamma_\omega(x)/4)^{2^m} \geq (\gamma_\omega(L)/4)^{2^m}$, and since $m \approx \log(n)/\log(\eta)$, the corollary follows. \square

Note that it is easy to prove Proposition 7 with the constant $\eta = 4$. Then by Corollary 8 we would obtain $\gamma_G \gtrsim e^{\sqrt{n}}$.

Proof of Proposition 7 for $\eta = 4$. Consider the weight $\omega(s) = 1$ for all $s \in S$ giving $\partial_\omega(g) = |g|$, and consider the following two homomorphisms (that they are homomorphisms was proven by Igor Lysionok [Lys85]):

$$\begin{aligned} \sigma : & \begin{cases} G \rightarrow H \\ a \mapsto c^a, & b \mapsto d, & d \mapsto c, & c \mapsto b; \end{cases} \\ \tau : & \begin{cases} G \rightarrow G \\ a \mapsto d, & b \mapsto 1, & d \mapsto a, & c \mapsto a. \end{cases} \end{aligned}$$

Then for any $g \in G$ we have $\psi(\sigma(g)) = (\tau(g), g)$, and $\tau(g)$ is of length at most 4, because $\tau(G) = \langle a, d \rangle$ is a dihedral group of order 8, and $|\sigma(g)| \leq 2|g| + 1$. Now for any $h \in H$, with $\psi(h) = (h_0, h_1)$, we have

$$h = a\sigma(h_0)a\sigma(\tau(h_0)^{-1}h_1);$$

indeed applying ψ to the right-hand side gives

$$(h_0, \tau(h_0))(\tau(\tau(h_0)^{-1}h_1), \tau(h_0)^{-1}h_1) = (h_0\tau(*), h_1) = (h_0, h_1),$$

because $(\tau(G), 1) \cap \psi(H) = (1, 1)$. Therefore we have

$$\begin{aligned} |h| &\leq |a| + |\sigma(h_0)| + |a| + |\sigma\tau(*)| + |\sigma(h_1)| \\ &\leq 1 + (2|h_0| + 1) + 1 + 8 + (2|h_1| + 1) \leq 4 \max\{|h_0|, |h_1|\} + 12. \end{aligned}$$

□

5. FINITE STATE AUTOMATA

We prove Proposition 7 by constructing a “finite state automaton” (see [HU79] for an introduction) that, given a pair (h_0, h_1) of words in $\psi(H)$, constructs a preimage h satisfying (1).

The automaton operates as follows: first each of the input words is extended in an infinite word by “padding”: extending it to the right by a padding symbol \div . At each step, the machine either outputs a letter, or reads and deletes the first letter on each of the input words. The concatenation of the output letters is the word “computed” by the machine.

Such a machine is most conveniently described as a directed, labelled graph Γ . The vertices (called *states*) of the graph correspond to internal states of the machine, and the edges (called *transitions*) to input or output. Vertices are of two types: *input* and *output*. The edges leaving an input state are called *input transitions* and are labelled by a pair (x, y) of input letters. The edges leaving an output state are called *output transitions* and are labelled by an output letter. There are two distinguished vertices, the *initial* and *final* states $*$ and \dagger .

Given a path $e = e_1 \dots e_n$ in Γ , we write $i(e) = (i(e)_0, i(e)_1)$ for the pair of words obtained by reading in sequence along e the labels of input transitions, and $o(e)$ for the word obtained by reading the labels of output transitions.

Let (u_0, u_1) be a pair of input words. Let e be a path in Γ from $*$ to \dagger . If $i(e)$ is of the form $(u_0 \div^*, u_1 \div^*)$, then $o(e)$ is a possible output of the automaton.

Note that the output may not be uniquely determined — if there is always at most one output, the automaton is called *deterministic*. Note also that in some cases the automaton fails to produce an output (if there exists no appropriate path). In our situation these problems will not occur.

We now add an important condition: that the graph Γ be finite. (This means that the automaton has a bounded amount of available memory.)

Lemma 9. *Let Γ be a finite state automaton, and let ∂ be a weight on (input and output) words. Assume that for every directed loop ℓ in Γ the following holds: $\partial o(\ell) < \eta/2(\partial i(\ell)_0 + \partial i(\ell)_1)$.*

Then there is a constant K such that, for every input pair (u_0, u_1) , there is an output word v with

$$\partial v \leq \eta \max\{\partial u_0, \partial u_1\} + K.$$

Proof. Let e be a path in Γ from $*$ to \dagger with $i(e) = (u_0 \div^*, u_1 \div^*)$. Since the graph is finite, say with N vertices, e may be decomposed as a product of paths $f_0 \ell_1 f_1 \dots \ell_m f_m$,

such that the ℓ_i are all loops in Γ and $f_0 f_1 \dots f_m$ is of length at most N . Let K be the maximal weight of a word of length at most N . Then

$$\begin{aligned} \partial v &= \sum_{i=1}^m \partial o(\ell_i) + \sum_{i=0}^m \partial o(f_i) \\ &\leq \sum_{i=1}^m \eta/2(\partial i(\ell)_0 + \partial i(\ell)_1) + K \\ &\leq \eta/2(\partial u_0 + \partial u_1) + K \leq \eta \max\{\partial u_0, \partial u_1\} + K. \end{aligned}$$

□

6. DESCRIPTION OF Γ

We now describe the graph Γ that computes ψ -preimages for the proof of Proposition 7. First we fix a minimal form \bar{g} for every $g \in G$, and denote by $\mathcal{M} \subset S^*$ the set of minimal forms. In Section 7 we will describe a computer-aided process that constructs a triangular weight $\omega : \{a, b, c, d\} \rightarrow \mathbb{R}_+^*$ and the finite oriented graph Γ .

Let us for now highlight the important properties of Γ :

- $V(\Gamma) \subset \mathcal{M} \times \mathcal{M}$, i.e. each vertex is identified with a pair of words (u_0, u_1) in minimal form. Using this identification, $V(\Gamma)$ is also viewed as a subset of $G \times G$.
The vertex set of Γ corresponds to the machine's "memory buffer": roughly, when it reads a symbol pair, it adds it at the right of its buffer, while when it outputs a symbol, it deletes some part on the left of its buffer.
- $* = \dagger = (\lambda, \lambda)$ is a vertex, where λ is the empty word.
- The definition in the previous section is slightly extended in that words (and not just letters) are allowed as edge-labels. Input transitions shall have labels of the form (xa, ya) for all $x, y \in \{b, c, d\}$. Output vertices shall have a unique outbound edge, with a label in \mathcal{M} .
- An input transition labelled (xa, ya) at a vertex (u_0, u_1) shall end at $(\overline{u_0 x a}, \overline{u_1 y a})$. An output transition labelled v at a vertex (u_0, u_1) shall end at $(\overline{v_0 u_0}, \overline{v_1 u_1})$.
- At every input vertex $(u_0, u_1) \in \psi(H)$, and for every $u \in \mathcal{M} \cap B$ of length of most 8, there is an input transition labelled $(\div^{|u|}, u)$ from (u_0, u_1) to a new vertex, directly followed by an output transition to (λ, λ) labelled by some v with $\psi(v) = (u_0, u_1 u)$.
- Except for these extra transitions, Γ satisfies Lemma 9 with the same constant η .

For now, we suppose such a graph exists, and prove the proposition under that assumption.

Proof of Proposition 7. Let $(u_0, u_1) \in \psi(H)$ be a pair of words in minimal form; we are to construct a word v such that $\psi(v) = (u_0, u_1)$ satisfies (1).

First we may suppose that neither u_0 nor u_1 starts by 'a'; indeed we may add b 's at their beginning (remembering that $(b, 1)$ and $(1, b)$ are in $\psi(H)$) and construct a word v' such that $\psi(v') = (b^i u_0, b^j u_1)$ with $i, j \in \{0, 1\}$. The word $v = (ada)^i d^j v'$ then satisfies $\psi(v) = (u_0, u_1)$ and is of weight at most 4 more than v' , a fact that can be coped with by increasing the constant K .

Second, we may suppose that u_0 is shorter than u_1 , and that their lengths differ by at most 8. Indeed the shorter of these two can be extended by copies of $(ad)^4$ which is trivial in G , and u_1 can be extended by an extra copy of $(ad)^4$. Again the cost of this operation is at worst an increase of K by 8.

Now Γ satisfies Lemma 9 for the constant η . This precisely means that there is an output word v satisfying (1). □

7. CONSTRUCTION OF Γ

In this section we explain how a computer may be programmed to construct the graph Γ described in the previous section. The construction itself involved a large amount of experimenting, to find adequate values for the constants described below.

First an arbitrary triangular weight is selected, for instance $\omega(s) \equiv 1$ (in practice, values between 0.9 and 1.1 work well); a tiny δ (about 0.01), a largish η' (around 4) and a limit-length N (around 20) are also chosen.

The *quality* of an output transition e labelled v from (u_0, u_1) to (u'_0, u'_1) is defined as

$$q(e) = \frac{\partial(u_0) + \partial(u_1) - \partial(u'_0) - \partial(u'_1)}{\partial v} + \delta |\partial(u_0) - \partial(u_1)| - \delta |\partial(u'_0) - \partial(u'_1)|.$$

Therefore, edges connecting a vertex of large total weight to a vertex of small total weight have a high quality. Edges connecting a vertex of dissimilar weight to a vertex of similar-weight words also have high quality.

The graph Γ is now constructed iteratively. At all steps of the iteration we shall have a graph Γ satisfying all the required conditions, except that it will have “hanging edges”, that is, edges not connected to any vertex. The purpose of the iteration process will then be to attach the hanging edges, either to existent vertices in Γ or to new ones.

The graph Γ starts with a single vertex, (λ, λ) , and a hanging edge. Then, while there are hanging edges, the following is performed:

- At each node (u_0, u_1) with a hanging edge, all words $v \in \mathcal{M} \cap H$ of length at most N are tried; we write $\psi(v) = (v_0, v_1)$.
- If the quality of the contracting edge from (u_0, u_1) to $(\overline{v_0 u_0}, \overline{v_1 u_1})$ is at least $1/\eta'$, then the type of (u_0, u_1) is set to “output”, and the hanging edge is replaced by an output transition from (u_0, u_1) to $(\overline{v_0 u_0}, \overline{v_1 u_1})$, labelled v .
- If there is no such output transition of sufficient quality, then the type of (u_0, u_1) is set to “input”, and the hanging edge is replaced by 9 edges to all the $(\overline{u_0 x a}, \overline{u_1 y a})$ for all $x, y \in \{b, c, d\}$.

If the process stops, we have obtained a graph Γ satisfying all constraints, including Lemma 9 for the constant η' . In fact, it may well satisfy Lemma 9 for some smaller value of η : the triangular weight may be varied, and it may happen that all edges of low quality are surrounded by edges of high quality.

For this purpose, a second program was written. It takes as input the description of Γ , and adjusts slightly the triangular weight ω ; then it searches Γ for loops, and on each loop computes the weight of the input- and output-labels. If the adjustment decreases the maximal ratio of input-weight to output-weight, it is kept. Then another adjustment is tried, etc.

Finally the “special transitions” $(\div^{|u|}, u)$ are added to Γ .

In actual computations, a graph Γ with 160 vertices (of which 12 are input states) was constructed. An appropriate weight was then chosen to be

$$\omega(a) = 1, \quad \omega(b) = 3.33, \quad \omega(c) = 2.8, \quad \omega(d) = 1.06.$$

The resulting η was 3.83414.

APPENDIX A. THE GRAPH Γ

In this printout, the graph Γ is described. The data are structured in “paragraphs”, each paragraph corresponding to a state of type “input”. Its nine successors are listed in the order $(da, da), (da, ca), \dots, (ba, ba)$, and each of these successors is identified by its type: if it is of type “output”, it is followed by the symbol

$$\text{OUTPUT}(w) \rightarrow (u_0, u_1)$$

meaning the word w is output and the next state to be in is (u_0, u_1) . If the successor is of type “input”, it is followed by the symbol **INPUT** and its successors are described in another paragraph.

Input states are listed in lexicographical order, starting with pairs (u_0, u_1) of same length, then pairs with length-difference 1, then 2, etc. In the lexicographical order d comes before c which comes before b . The input states are (λ, λ) , (da, da) , (da, ca) , (ca, da) , (ca, ca) , (a, λ) , (da, d) , (da, λ) , (ca, λ) , (ad, λ) , (ada, λ) and (aca, λ) .

$$\begin{array}{ll}
(\lambda, \lambda): & \text{INPUT} \rightarrow (\text{da}, \text{da}): \text{INPUT} \\
& (\text{da}, \text{ca}): \text{INPUT} \\
& (\text{da}, \text{ba}): \text{OUTPUT}(\text{acad}) \rightarrow (\text{a}, \lambda) \\
& (\text{ca}, \text{da}): \text{INPUT} \\
& (\text{ca}, \text{ca}): \text{INPUT} \\
& (\text{ca}, \text{ba}): \text{OUTPUT}(\text{abad}) \rightarrow (\text{a}, \lambda) \\
& (\text{ba}, \text{da}): \text{OUTPUT}(\text{cada}) \rightarrow (\text{a}, \lambda) \\
& (\text{ba}, \text{ca}): \text{OUTPUT}(\text{bada}) \rightarrow (\text{a}, \lambda) \\
& (\text{ba}, \text{ba}): \text{OUTPUT}(\text{abad}) \rightarrow (\text{da}, \lambda)
\end{array}$$
$$(da, da): \text{ INPUT} \rightarrow (dada, dada) = (adad, adad): \text{ OUTPUT}(cacacaca) \rightarrow (\lambda, \lambda)$$
$$\begin{array}{l}
\text{(dada,daca)=(adad,daca):} \quad \text{OUTPUT(cacabaca)} \rightarrow (\lambda,\lambda) \\
\text{(dada,daba)=(adad,daba):} \quad \text{OUTPUT(cacab)} \rightarrow (\text{da},\text{d}) \\
\text{(daca,dada)=(daca,adad):} \quad \text{OUTPUT(acacabac)} \rightarrow (\lambda,\lambda) \\
\text{(daca,daca):} \quad \text{OUTPUT(acabacadadab)} \rightarrow (\lambda,\lambda) \\
\text{(daca,daba):} \quad \text{OUTPUT(cabacaca)} \rightarrow (\text{aca},\lambda) \\
\text{(daba,dada)=(daba,adad):} \quad \text{OUTPUT(acacaba)} \rightarrow (\text{da},\text{d}) \\
\text{(daba,daca):} \quad \text{OUTPUT(acabacac)} \rightarrow (\text{aca},\lambda) \\
\text{(daba,daba):} \quad \text{OUTPUT(acabacacabacaca)} \rightarrow (\text{ad},\lambda) \\
\text{(da,ca):} \quad \text{INPUT} \rightarrow \text{(dada,cada)=(adad,cada):} \quad \text{OUTPUT(bacacaca)} \rightarrow (\lambda,\lambda) \\
\text{(dada,caca)=(adad,caca):} \quad \text{OUTPUT(bacabaca)} \rightarrow (\lambda,\lambda) \\
\text{(dada,caba)=(adad,caba):} \quad \text{OUTPUT(bacab)} \rightarrow (\text{da},\text{d}) \\
\text{(daca,cada):} \quad \text{OUTPUT(cabacacad)} \rightarrow (\lambda,\lambda)
\end{array}$$

```

(data, caca):    OUTPUT(acababab) → (λ, λ)
(daca, caba):    OUTPUT(cabacacada) → (aca, λ)
(daba, cada):    OUTPUT(cacadaba) → (da, d)
(daba, caca):    OUTPUT(acabacab) → (aca, λ)
(daba, caba):    OUTPUT(cabacacabacab) → (ad, λ)
(ca, da):  INPUT → (cada, dada)=(cada, adad): OUTPUT(abacacac) → (λ, λ)
(cada, daca):    OUTPUT(acabacacada) → (λ, λ)
(cada, daba):    OUTPUT(acacadaab) → (da, d)
(caca, dada)=(caca, adad): OUTPUT(abacabac) → (λ, λ)
(caca, daca):    OUTPUT(cabacababadab) → (λ, λ)
(caca, daba):    OUTPUT(cabacaba) → (aca, λ)
(caba, dada)=(caba, adad): OUTPUT(abacaba) → (da, d)
(caba, daca):    OUTPUT(acabacacada) → (aca, λ)
(caba, daba):    OUTPUT(acabacacabacaba) → (ad, λ)
(ca, ca):  INPUT → (cada, cada): OUTPUT(acacacabada) → (λ, λ)
(cada, caca):    OUTPUT(acabacabada) → (λ, λ)
(cada, caba):    OUTPUT(acabadab) → (da, d)
(caca, cada):    OUTPUT(cabacabad) → (λ, λ)
(caca, caca):    OUTPUT(acabacabadabadab) → (λ, λ)
(caca, caba):    OUTPUT(cabacabad) → (aca, λ)
(caba, cada):    OUTPUT(cabadaba) → (da, d)
(caba, caca):    OUTPUT(acabacabada) → (aca, λ)
(caba, caba):    OUTPUT(badabababadababac) → (ad, λ)

```

$$\begin{array}{lll} (a, \lambda): & \text{INPUT} & \rightarrow (ada, da): \text{OUTPUT}(caca) \rightarrow (a, \lambda) \\ & & (ada, ca): \text{OUTPUT}(baca) \rightarrow (a, \lambda) \\ & & (ada, ba): \text{OUTPUT}(b) \rightarrow (da, da) \\ & & (aca, da): \text{OUTPUT}(caba) \rightarrow (a, \lambda) \\ & & (aca, ca): \text{OUTPUT}(baba) \rightarrow (a, \lambda) \\ & & (aca, ba): \text{OUTPUT}(b) \rightarrow (ca, da) \\ & & (aba, da): \text{OUTPUT}(caba) \rightarrow (da, \lambda) \end{array}$$

$(aba, ca): \text{OUTPUT}(baba) \rightarrow (da, \lambda)$
 $(aba, ba): \text{OUTPUT}(badac) \rightarrow (a, \lambda)$
 $(da, d): \text{INPUT} \rightarrow (dada, dda)=(adad, a): \text{OUTPUT}(aca) \rightarrow (ada, \lambda)$
 $(dada, dca)=(adad, ba): \text{OUTPUT}(acad) \rightarrow (ada, \lambda)$
 $(dada, dba)=(adad, ca): \text{OUTPUT}(baca) \rightarrow (ad, \lambda)$
 $(daca, dda)=(daca, a): \text{OUTPUT}(aca) \rightarrow (aca, \lambda)$
 $(daca, dca)=(daca, ba): \text{OUTPUT}(acad) \rightarrow (aca, \lambda)$
 $(daca, dba)=(daca, ca): \text{OUTPUT}(cabacacad) \rightarrow (ad, \lambda)$
 $(daba, dda)=(daba, a): \text{OUTPUT}(acabadab) \rightarrow (\lambda, \lambda)$
 $(daba, dca)=(daba, ba): \text{OUTPUT}(acacadab) \rightarrow (\lambda, \lambda)$
 $(daba, dba)=(daba, ca): \text{OUTPUT}(badabaca) \rightarrow (aca, \lambda)$

now come vertices whose lengths differ by 2:

$(da, \lambda): \text{INPUT} \rightarrow (dada, da)=(adad, da): \text{OUTPUT}(caca) \rightarrow (ad, \lambda)$
 $(dada, ca)=(adad, ca): \text{OUTPUT}(baca) \rightarrow (ad, \lambda)$
 $(dada, ba)=(adad, ba): \text{OUTPUT}(acad) \rightarrow (ada, \lambda)$
 $(daca, da): \text{OUTPUT}(cabacaca) \rightarrow (ad, \lambda)$
 $(daca, ca): \text{OUTPUT}(cabacacad) \rightarrow (ad, \lambda)$
 $(daca, ba): \text{OUTPUT}(acad) \rightarrow (aca, \lambda)$
 $(daba, da): \text{OUTPUT}(badabaca) \rightarrow (ada, \lambda)$
 $(daba, ca): \text{OUTPUT}(badabaca) \rightarrow (aca, \lambda)$
 $(daba, ba): \text{OUTPUT}(acacadab) \rightarrow (\lambda, \lambda)$
 $(ca, \lambda): \text{INPUT} \rightarrow (cada, da): \text{OUTPUT}(acacada) \rightarrow (ad, \lambda)$
 $(cada, ca): \text{OUTPUT}(acabada) \rightarrow (ad, \lambda)$
 $(cada, ba): \text{OUTPUT}(abad) \rightarrow (ada, \lambda)$
 $(caca, da): \text{OUTPUT}(cabacaba) \rightarrow (ad, \lambda)$
 $(caca, ca): \text{OUTPUT}(cabacabad) \rightarrow (ad, \lambda)$
 $(caca, ba): \text{OUTPUT}(abad) \rightarrow (aca, \lambda)$
 $(caba, da): \text{OUTPUT}(badababa) \rightarrow (ada, \lambda)$
 $(caba, ca): \text{OUTPUT}(badababa) \rightarrow (aca, \lambda)$
 $(caba, ba): \text{OUTPUT}(abacadab) \rightarrow (\lambda, \lambda)$
 $(ad, \lambda): \text{INPUT} \rightarrow (adda, da)=(\lambda, da): \text{INPUT}$

in fact, the destination edge is (da, λ) , since we list edge pairs with longest word first. This is irrelevant, since word reductions are similar when operating on (u_0, u_1) and on (u_1, u_0)

$(adda, ca)=(\lambda, ca): \text{INPUT}$
 $(adda, ba)=(\lambda, ba): \text{OUTPUT}(ada) \rightarrow (a, \lambda)$
 $(adca, da)=(aba, da): \text{OUTPUT}(caba) \rightarrow (da, \lambda)$
 $(adca, ca)=(aba, ca): \text{OUTPUT}(baba) \rightarrow (da, \lambda)$
 $(adca, ba)=(aba, ba): \text{OUTPUT}(badac) \rightarrow (a, \lambda)$
 $(adba, da)=(aca, da): \text{OUTPUT}(caba) \rightarrow (a, \lambda)$
 $(adba, ca)=(aca, ca): \text{OUTPUT}(baba) \rightarrow (a, \lambda)$
 $(adba, ba)=(aca, ba): \text{OUTPUT}(b) \rightarrow (ca, da)$

finally some vertices whose lengths differ by 3:

$(ada, \lambda): \text{INPUT} \rightarrow (adada, da)=(dad, da): \text{OUTPUT}(acac) \rightarrow (ada, \lambda)$
 $(adada, ca)=(dad, ca): \text{OUTPUT}(acab) \rightarrow (ada, \lambda)$
 $(adada, ba)=(dad, ba): \text{OUTPUT}(acad) \rightarrow (ad, \lambda)$
 $(adaca, da): \text{OUTPUT}(c) \rightarrow (daca, a)$
 $(adaca, ca): \text{OUTPUT}(b) \rightarrow (daca, a)$
 $(adaca, ba): \text{OUTPUT}(b) \rightarrow (daca, da)$
 $(adaba, da): \text{OUTPUT}(c) \rightarrow (daba, a)$
 $(adaba, ca): \text{OUTPUT}(b) \rightarrow (daba, a)$
 $(adaba, ba): \text{OUTPUT}(b) \rightarrow (daba, da)$
 $(aca, \lambda): \text{INPUT} \rightarrow (acada, da): \text{OUTPUT}(caba) \rightarrow (ada, \lambda)$
 $(acada, ca): \text{OUTPUT}(baba) \rightarrow (ada, \lambda)$
 $(acada, ba): \text{OUTPUT}(acacadab) \rightarrow (ad, \lambda)$
 $(acaca, da): \text{OUTPUT}(caba) \rightarrow (aca, \lambda)$
 $(acaca, ca): \text{OUTPUT}(baba) \rightarrow (aca, \lambda)$
 $(acaca, ba): \text{OUTPUT}(b) \rightarrow (caca, da)$
 $(acaba, da): \text{OUTPUT}(cababadab) \rightarrow (\lambda, \lambda)$
 $(acaba, ca): \text{OUTPUT}(bababadab) \rightarrow (\lambda, \lambda)$
 $(acaba, ba): \text{OUTPUT}(b) \rightarrow (caba, da)$

REFERENCES

- [Bar98] Laurent Bartholdi, *The growth of Grigorchuk's torsion group*, Internat. Math. Res. Notices **20** (1998), 1349–1356.
- [BG98] Laurent Bartholdi and Rostislav I. Grigorchuk, *Lie methods in growth of groups and groups of finite width*, to appear, 1998.
- [GH90] Étienne Ghys and Pierre de la Harpe, *Sur les groupes hyperboliques d'après Mikhael Gromov*, Progress in Mathematics, vol. 83, Birkhäuser Boston Inc., Boston, MA, 1990, Papers from the Swiss Seminar on Hyperbolic Groups held in Bern, 1988.
- [Gri80] Rostislav I. Grigorchuk, *On Burnside's problem on periodic groups*, Funct. Anal. Appl. **14** (1980), 41–43.
- [Gri83] Rostislav I. Grigorchuk, *On the Milnor problem of group growth*, Dokl. Akad. Nauk SSSR **271** (1983), no. 1, 30–33.
- [Gri85] Rostislav I. Grigorchuk, *Degrees of growth in finitely generated groups, and the theory of invariant means*, Math. USSR-Izv. **25** (1985), no. 2, 259–300.
- [Gri91] Rostislav I. Grigorchuk, *On growth in group theory*, Proceedings of the International Congress of Mathematicians, Kyoto, 1990 (Tokyo), vol. I, Math. Soc. Japan, 1991, pp. 325–338.
- [Gro81] Mikhael Gromov, *Groups of polynomial growth and expanding maps*, Inst. Hautes Études Sci. Publ. Math. (1981), no. 53, 53–73.
- [Har] Pierre de la Harpe, *Topics in geometric group theory*, University of Chicago Press, to appear; preprint at <http://www.unige.ch/math/biblio/preprint/pp98.html>.
- [HU79] John E. Hopcroft and Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley Series in Computer Science, Addison-Wesley Publishing Co., Reading, Mass., 1979.
- [Leo98] Yuriĭ G. Leonov, *On growth function for some torsion residually finite groups*, International Conference dedicated to the 90th Anniversary of L.S. Pontryagin (Moscow), vol. Algebra, Steklov Mathematical Institute, September 1998, pp. 36–38.
- [Lys85] Igor G. Lysionok, *A system of defining relations for the Grigorchuk group*, Mat. Zametki **38** (1985), 503–511.
- [Mil68a] John W. Milnor, *Growth of finitely generated solvable groups*, J. Differential Geom. **2** (1968), 447–449.
- [Mil68b] John W. Milnor, *Problem 5603*, Amer. Math. Monthly **75** (1968), 685–686.
- [MP99] Roman Muchnik and Igor Pak, *On growth of Grigorchuk groups*, preprint, 1999.
- [Sva55] A. S. Svarts, *A volume invariant of coverings*, Dokl. Akad. Nauk SSSR (1955), no. 105, 32–34 (Russian).
- [Tit72] Jacques Tits, *Free subgroups in linear groups*, J. Algebra **20** (1972), 250–270.

E-mail address: Laurent.Bartholdi@math.unige.ch

SECTION DE MATHÉMATIQUES
UNIVERSITÉ DE GENÈVE
CP 240, 1211 GENÈVE 24
SWITZERLAND